

Appendix H – Overview of Quality Control Tools/Features

1.0 General Information

This section has been written specifically for build ob7.2 and is an overview of quality control tools/features that are available in this build. It includes information on database tables that contain quality control information, and provides information about which RAX applications use these tables.

2.0 Database tables

2.1 *datalimits* and *locdatalimits* tables

The *datalimits* and *locdatalimits* tables are based on the tables with the same names in the IHFS database. Information from the IHFS database can easily be grabbed and with some minor modifications become load files for the RFC Archive Database versions of these tables. Currently the only quality controls fields that are utilized by any of the RAX baseline applications are: *gross_range_min*, *gross_range_max*, *reason_range_min* and *reason_range_max*.

2.2 *sensok* table

The *sensok* table allows the user to manually specify periods of bad 'raw' data. This table is read by the level 1 processor applications. For detailed information see section L-2.

2.3 *rivercrit* table

While the *rivercrit* table is not exclusively for quality control purposes, it does contain fields that allow the user to define some quality controls that are used by the level 1 processors. These fields are: *lowscreen*, *sigrate*, *screenrate*, *highscreen*, *damscreen*, *lowscreenf*, *sigratef*, *screenratef*, *highscreenf*, *damscreenf*, *sigratet*, *screenratet*, *lowscreenq*, *sigrateq*, *screenrateq*, *highscreenq*, and *damscreenq*. See Appendix G for more information about these fields.

3.0 Applications

3.1 *shef_decode_raw* and *shef_decode_pro*

Just like the IHFS *shefdecoder*, these *shefdecoders* will perform some simple quality control tests based on information that is stored in the *datalimits* and *locdatalimits*

tables. If the user has defined information in one or both of these tables the decoders will perform the test(s) and compute a `quality_code` value. For psuedo array SHEF data value tables where there is no `quality_code` field and only a SHEF qualifier code, the value is converted to “G”, “Q”, “B” or “Z” accordingly. For single value per row SHEF data values tables the QC information is stored both ways. For generalized information about the quality code value and how it is computed, see the quality code guide at https://ocwws.weather.gov/intranet/whfs/documentation/IHFS_qcops_OB3.doc.

3.2 level 1 processors

The level 1 processors use the SHEF qualifier code fields in the “raw” SHEF data value tables and the *sensok* table. For more information see sections G-2 and L-2.

3.3 DatView

At this time DatView only allows the viewer to view the SHEF qualifier code and quality code data if that field exists in the SHEF data value tables. The user CAN NOT filter the data being requested using the SHEF qualifier code or quality code fields.

3.4 National River Verification

In build ob7.2, the national river verification software, IVP, uses the SHEF data value tables SHEF qualifier code fields during the pairing process. It has a hardwired, ordered string that defines a quality code preference. The order is: G, M, P, V, S, Z, T, F, Q. The pairing process will never use observed data that is flagged with B, R or E. The observed data value's SHEF qualifier code is the first item checked when making a pair. So, if two observed values are found that could match the forecast value, and one has a quality code of 'Z' and the other has 'B', the one with 'Z' will be paired with the forecast. This is regardless of the sensor preference and validtime-obstime difference (except that the difference must still be within the pairing window).

If, at a later time, the 'B' forecast is examined, fixed, and a new record is put back into the archive database with a 'G', and the pairing algorithm is executed again for the original forecast, then a new pair will be constructed with a quality code of 'G' using the adjusted observed value. This will replace the pair already in the *vfypairs* table, as the prime key of the existing pair will match that of the newly constructed pair.